

Escuela Politécnica Superior

20  
21

# Trabajo fin de grado

Desarrollo de una aplicación web para reserva y pago en pequeño comercio.



Sergio Arranz González

Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
C/ Francisco Tomás y Valiente nº 11



**UNIVERSIDAD AUTÓNOMA DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR**



**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Desarrollo de una aplicación web para reserva y  
pago en pequeño comercio.**

**Autor: Sergio Arranz González  
Tutor: Alberto Suárez**

**junio 2021**

#### **Algunos derechos reservados.**

Este trabajo está bajo licencia Creative Commons

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Esta obra se puede copiar, distribuir y comunicar públicamente la obra así como crear obras derivadas bajo las siguientes condiciones:

- Debe reconocer los créditos manteniendo la autoría original y añadiendo la autoría de las modificaciones indicando de forma expresa y bien visible que el autor original no manifiesta ningún tipo de apoyo a las modificaciones realizadas así como al uso que se da de esta obra.
- No se puede utilizar esta obra con fines comerciales.
- Las modificaciones o ediciones de esta obra deben compartirse bajo una licencia idéntica a esta.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

#### **DERECHOS RESERVADOS**

© 3 de Noviembre de 2021 por UNIVERSIDAD AUTÓNOMA DE MADRID

Francisco Tomás y Valiente, nº 1

Madrid, 28049

Spain

**Sergio Arranz González**

**Desarrollo de una aplicación web para reserva y pago en pequeño comercio.**

**Sergio Arranz González**

C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

*A mi familia y mi novia*

*En los momentos de crisis,  
solo la imaginación es más importante que el conocimiento*

*Albert Einstein*



# AGRADECIMIENTOS

---

Primeramente me gustaría agradecer a Alberto Suárez González, por toda la ayuda prestada y por ofrecerme esta gran oportunidad de desarrollar mis conocimientos adquiridos para ayudar a los demás.

Finalmente agradecer también a Ander Gómez Juaristi, dueño de Panifiesto, por colaborar y embarcarse en esta aventura que supone la digitalización de su negocio.





# RESUMEN

---

Este proyecto surge en el contexto de una pandemia mundial causada por la COVID-19 y de las medidas adoptadas para prevenir contagios derivados de aglomeraciones. Las principales medidas son el uso de mascarillas, el aumento de la distancia social y reducción de aforos. Es esta última es la que más afecta a pequeños comercios al ver limitada su capacidad de atender a los clientes con la misma agilidad que en la época prepandemia, con el resultado de demoras y otras deficiencias de calidad en el servicio proporcionado al cliente.

Este Trabajo de fin de grado (TFG) es una oportunidad para mejorar la calidad del servicio proporcionado al cliente en pequeños comercios en las condiciones adoptadas para dar respuesta a la emergencia sanitaria. Uno de estos comercios es Panifiesto, una panadería agroecológica situada en Madrid centro, en la cual por la reducción de aforo, no es posible atender a varios clientes de manera simultánea. En este sentido, el principal propósito es diseñar e implementar una aplicación web que permita reservar de antemano los productos a adquirir. Así mismo, esta aplicación está diseñada de forma que se puedan integrar sistemas de pago electrónico. De esta forma se reducirá el tiempo de presencia física del cliente en el local del comercio, con la consiguiente mejora del servicio. Adicionalmente, la aplicación facilita la planificación para dar respuesta a la demanda esperada. Permite también recoger datos sobre las transacciones de forma que se puedan aplicar algoritmos de aprendizaje para mejorar la gestión del negocio.

Si bien se ha tomado el caso concreto de una tahona con el fin de abordar el diseño de la aplicación con la participación de un usuario real, la aplicación puede ser configurada y adaptada al perfil de otros pequeños comercios con problemas similares.

Tras la pandemia causada por la Covid19 algunos comercios deben adaptarse para mantener sus negocios a flote mientras cumplen con las nuevas normativas de seguridad.

El objetivo de este trabajo de fin de grado es ofrecer una solución para uno de estos comercios mediante la elaboración de una herramienta que permita la realización y gestión de reservas junto con un sistema de pago para agilizar los procesos de adquisición de productos.

# PALABRAS CLAVE

---

Aplicación web, Digitalización, Pequeño comercio, Gestión de reservas, Pagos en remoto



# ABSTRACT

---

This project emerges in the context of a global pandemic caused by COVID-19 and the measures taken to prevent contagion from crowds. The main measures are the use of masks, increased social distance and capacity reduction. The latter is the one that most affects small shops as their ability to serve customers with the same agility as in the pre-pandemic period is limited, resulting in delays and other quality deficiencies in the service provided to the customer.

This Final Degree Project (FDP) is an opportunity to improve the quality of service provided to customers in small shops under the conditions adopted to respond to the health emergency. One of these shops is Panifiesto, an agro-ecological bakery located in the centre of Madrid, in which, due to the reduced capacity, it is not possible to serve several customers simultaneously. In this sense, the main purpose is to design and implement a web application that allows to book in advance the products to be purchased. This application is also designed in such a way that electronic payment systems can be integrated. This will reduce the time the customer has to be physically present in the shop, thus improving the service. In addition, the application facilitates planning to meet expected demand. It also makes it possible to collect data on transactions so that learning algorithms can be applied to improve business management.

Although the specific case of a bakery has been taken in order to approach the design of the application with the participation of a real user, the application can be configured and adapted to the profile of other small businesses with similar problems.

After the pandemic caused by Covid19 some businesses must adapt to keep their business afloat while complying with the new security regulations.

The goal of this final degree project is to offer a solution for one of these businesses by developing a tool that allows for the creation and management of reservations together with a payment system to speed up product acquisition processes.

# KEYWORDS

---

Web app, Digitalisation, Small commerce, Booking management, Remote payments



# ÍNDICE

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Estado del arte</b>	<b>3</b>
2.1	Amazon .....	3
2.2	Alibaba .....	4
2.3	TheFork y TripAdvisor .....	4
2.4	Too Good To Go .....	6
2.5	Conclusión del estado del arte .....	9
<b>3</b>	<b>Análisis</b>	<b>11</b>
3.1	Objetivos .....	11
3.2	Catálogo inicial de requisitos .....	12
<b>4</b>	<b>Diseño</b>	<b>19</b>
<b>5</b>	<b>Implementación</b>	<b>31</b>
5.1	Python .....	31
5.2	React y Angular .....	31
5.3	App nativa o Progressive Web App .....	31
5.4	Firebase .....	32
5.5	Javascript vs Typescript .....	32
5.6	Tecnologías a utilizar .....	32
5.7	Ciclo de vida del software .....	33
<b>6</b>	<b>Pruebas</b>	<b>35</b>
<b>7</b>	<b>Conclusiones</b>	<b>37</b>
<b>8</b>	<b>Trabajo futuro</b>	<b>39</b>
	<b>Bibliografía</b>	<b>41</b>
	<b>Acrónimos</b>	<b>43</b>



# LISTAS

---

## Lista de figuras

2.1	Opciones de compra en Amazon .....	4
2.2	Precios por unidades en Alibaba .....	5
2.3	Envío sin asignar Alibaba .....	5
2.4	Información de restaurantes en The Fork .....	6
2.5	Información de restaurantes en TripAdvisor .....	7
2.6	Listado de packs en Too Good To Go .....	8
3.1	Casos de uso del sistema .....	17
4.1	Menú de navegación de usuarios .....	20
4.2	Pantallas de control de usuarios .....	20
4.3	Pantallas de contacto y catálogo de productos .....	21
4.4	Pantalla de cesta .....	22
4.5	Menú de usuario .....	23
4.6	Historial de reservas del usuario .....	23
4.7	Pantalla de detalles del pedido .....	24
4.8	Pantalla de actualizar datos .....	24
4.9	Menú de administración .....	24
4.10	Pantalla de panes .....	25
4.11	Pantallas de gestión de panes .....	26
4.12	Listado de reservas .....	26
4.13	Añadir reserva .....	27
4.14	Editar reserva .....	28
4.15	Editar horarios de reserva .....	29
5.1	Diagrama de Gantt .....	34





# INTRODUCCIÓN

---

En el último año hemos vivido una situación que ha hecho cambiar la forma de relacionarnos con el mundo e interactuar con aquello que nos rodea. Hace años nadie hubiera imaginado una situación como la actual, con aforos en comercios y minimización del contacto físico entre los clientes y los productos que van a adquirir. Por este motivo, los comercios deben adaptarse. Esta adaptación pasa por encontrar nuevas vías que faciliten la compra de sus productos, evitando superar los aforos estipulados, reduciendo el contacto físico al mínimo y tratando de mantener sus ventas o incluso hacer crecer su negocio.

La digitalización del negocio mediante la implantación de aplicaciones informáticas que permitan una gestión más eficaz y eficiente y mejoren la interacción tanto remota como presencial con los clientes, permite dar solución a estos problemas respetando las recomendaciones sanitarias derivadas de la Covid19.

La funcionalidad de esta aplicación es común con sistemas de reservas en red para restaurantes, hoteles, productos de consumo, etc. Algunos ejemplos pueden ser 'El Tenedor', 'Trivago', 'Too Good To Go', etc. No obstante el sistema desarrollado permite la reserva de varios productos a la vez, seleccionar la hora que más convenga al cliente, modificar la reserva y recibir información sobre pedidos realizados.

El objetivo, con este proyecto, es elaborar una herramienta que permita la realización de reservas y el pago anticipado. Esta permitirá, a los usuarios propietarios de un comercio, gestionar sus horarios de reservas, el catálogo de productos que quieran añadir y gestionar los pedidos que tengan, de manera autónoma, sin necesidad de añadir nuevas funcionalidades.

Además mediante el uso de filtros por día de la semana, permitirá seleccionar tan solo los productos disponibles para ese día, sin dejar de mostrar el resto de productos.

Para cubrir todas estas funcionalidades se deberá manejar el sistema mediante una parte **frontal**, la cual muestre los datos en tiempo real y permita la interacción del usuario a través de ellos y una parte **trasera** encargada de gestionar, almacenar y servir todos los datos necesarios para el correcto funcionamiento de la aplicación. Además se ha elaborado teniendo en cuenta todas fases de un proyecto de software:

- En el capítulo 2 se analizan otras aplicaciones del mercado comparándolas con la aplicación que se quiere desarrollar.
- En el capítulo 3 se extraen los requisitos que busca un pequeño negocio a la hora de digitalizar su negocio de venta de productos y se elaboran casos de uso que cumplan esos requisitos.
- En el capítulo 4 se muestran maquetas que cumplen con los casos de uso establecidos previamente en el análisis.
- En el capítulo 5 se habla de las diferentes tecnologías que se tuvieron en cuenta inicialmente y cuales se utilizarán finalmente para el desarrollo.
- En el capítulo 6 se explica qué pruebas se han realizado durante el proceso de desarrollo y tras la finalización del desarrollo.
- En el capítulo 7 se extraen las conclusiones tras el desarrollo.
- En el capítulo 8 se evalúan los requisitos que no se han logrado satisfacer completamente y se exponen posibles mejoras futuras para la aplicación.

La herramienta obtenida será un prototipo inicial al que ir añadiendo mejoras y adaptaciones que permitan encajar con el negocio a digitalizar.

## ESTADO DEL ARTE

---

Para crear una aplicación de reservas, es importante tener en consideración servicios similares que ya existan para poder ver sus ventajas y sus defectos, de cara a poder seleccionar los puntos fuertes y mejorar los puntos débiles al elaborar nuestra aplicación.

Para ello, se ha realizado el estudio con varias aplicaciones que ya existen en el mercado y que se exponen a continuación.

### 2.1. Amazon

Una de las empresas pioneras y con gran presencia en el mercado es el sistema de compra online de Amazon. Actualmente existen numerosas aplicaciones que compiten por el mismo mercado como pueden ser El Corte Inglés, FNAC, etc. Estas empresas no solo ofrecen productos comercializados directamente por ellas, sino que también actúan como intermediarios para otros proveedores. Vamos a centrarnos en Amazon, ya que es la más veterana de todas y la más relevante a nivel mundial. Como se puede observar en la figura 2.1 ofrece dos opciones:

- 1.— Añadir a la cesta
- 2.— Comprar ya

La 1ª permite añadir más artículos a la cesta, mientras que la 2ª realizará el pedido automáticamente siempre que tengamos una tarjeta guardada y unos datos de envío por defecto.

Algunas características a tener en cuenta de Amazon son:

- Necesita datos personales del usuario para las entregas.
- Es necesario registrarse para realizar un pedido.
- Ofrece características extra, como envíos más rápidos o envíos gratuitos para usuarios que paguen una suscripción a la plataforma.
- Permite recibir el pedido en tu domicilio indicando una franja horaria y un día.
- Permite recoger el pedido en algún punto de entrega.
- Permite a los vendedores, dar de alta nuevos artículos, cada uno con sus características propias.
- Los vendedores externos, tienen que adaptarse a la política de envíos de Amazon, por lo que tienen menor autonomía.

Aun que Amazon comparte algunas características con nuestra aplicación, algo que las diferencia es que para nuestra aplicación todo lo que se reserva precisa de una interacción física con el comercio que vende los productos, mientras que en Amazon la recogida no implica contacto físico con el vendedor, tan solo con un repartidos y en caso de recogida en buzón, ni siquiera esto último.

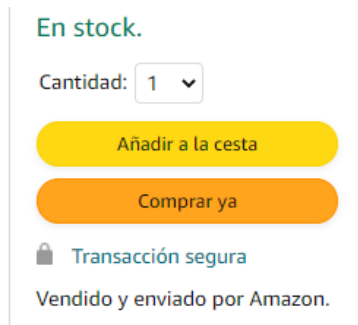


Figura 2.1: Opciones de compra en Amazon


## 2.2. Alibaba

Otra aplicación dedicada al comercio de productos mundial es Alibaba, originaria de China, supo expandirse por el resto del mundo ofreciendo a sus vendedores una plataforma con exposición mundial. Su principal diferencia respecto a Amazon, es que Alibaba aloja a vendedores en su plataforma, estos suben sus productos, permitiendo su compra al por mayor, convirtiéndola más en una plataforma intermediaria que en una tienda. El precio de los productos puede variar dependiendo del número de unidades que se compre, como puede observarse en la figura 2.2. Además, el envío dependerá del proveedor, pudiendo asignarse previamente como en el caso de la figura 2.2 o negociarse con el vendedor tras realizar el pedido como puede verse en 2.3, lo cual ofrece mayor libertad a los vendedores, pero supone un mayor esfuerzo por parte del cliente.

Con Alibaba hay un mayor parecido respecto al contacto con el vendedor, aun que en Alibaba es de manera online. También existe otra diferencia importante, ya que nuestra aplicación permite reservar productos en un comercio y Alibaba es una aplicación que hace de intermediaria entre el fabricante y el consumidor.

## 2.3. TheFork y TripAdvisor

A la hora de realizar la reserva de productos, también es importante tener en cuenta webs que permitan la reserva en restaurantes, como **TheFork**, antiguamente llamada El Tenedor. Es una de las principales webs de reservas online para restaurantes. Como podemos ver en la figura 2.4, The Fork proporciona información resumida que puede resultar útil

ras y abrigos de hombre  [Suscribirse a Comercio alerta](#) Obtener la aplicación | Español - EUR ▾

istrador si se trata de un producto de protección contra virus y si el coronavirus (COVID-19) ✕

**Listo para enviar** 🔒 En stock 🚚 Envío rápido

TONGYANG-Chaqueta de algodón para hombre, abrigo con cuello vuelto, gabardina entallada, novedad de invierno

★★★★☆ 4.0 1 Reseñas 10 compradores

1 - 10 Unidades	11 - 29 Unidades	30 - 49 Unidades	>=50 Unidades
13,25 €	12,01 €	11,18 €	10,35 €

11 Unidades **132,07 €**

Enviar a Spain por HK **85,25 €**

UPS Expedited

Tiempo de procesamiento **15 días** ⓘ

Tiempo de envío **6-10 días** ⓘ

---

**Total** **217,31 €**

Iniciar pedido

Figura 2.2: Precios por unidades en Alibaba

2 Unidades **312,99 €**

Enviar a Spain Para negociar

Tiempo de procesamiento **7 días** ⓘ

---

**Total**

**To Be Negotiated**

Iniciar pedido

✉ Contactar Proveedor

🛒 Añadir al carro de compras

Figura 2.3: Envío sin asignar Alibaba

para el visitante, como pueden ser la ubicación, el precio promedio y la valoración media del restaurante. Además de esto, es interesante saber que para la realización de la reserva deben los usuarios aportar:

- Número de comensales
- Día de la reserva
- Si será desayuno, cena o comida
- Hora para la reserva
- Correo electrónico, en caso de no estar registrados en la plataforma

**TripAdvisor** es similar a TheFork, pero con un catálogo más amplio de productos: reserva en hoteles, actividades y vuelos, además de restaurantes. Aunque su catálogo de servicios es más amplio, ofrece información menos detallada que la web TheFork como podemos observar en la figura 2.5. Para realizar reservas, en algunos casos solo podrá realizarse pagando por adelantado.

Nuestra aplicación comparte características con TheFork o Tripadvisor, ya que son aplicaciones orientadas a la reserva de productos pudiendo ser pagados previamente. Pero existe una diferencia fundamental y es que en estas aplicaciones solo se puede incluir un producto en cada reserva, mientras que en nuestra aplicación se podrán reservar todos los panes que el cliente necesite en tan solo una reserva.



Figura 2.4: Información de restaurantes en The Fork

## 2.4. Too Good To Go

Finalmente, la aplicación de reserva de productos excedentes en comercios, Too Good To Go. En dicha aplicación, las tiendas ponen a disposición de los usuarios un lote de productos, establecen un precio y estipulan una hora de recogida. Para acceder a la aplicación como consumidor los usuarios deben crear una cuenta. Una vez están dentro, deben elegir una zona en la que ver comercios cercanos que ofrezcan packs de productos. Tras elegir zona y radio de acción, aparecerán los comercios con los packs disponibles, su precio y su hora de recogida como puede verse en la figura 2.6. Los usuarios



**Figura 2.5:** Información de productos en TripAdvisor

reservan los productos seleccionando el comercio y pagan el pack de productos desde la aplicación una vez se recoja en el establecimiento, pero no podrán elegir hora a la que recogerlo, ya que viene prefijada por el comercio del que se recoge.

Con esta aplicación compartimos el contacto con los clientes, ya que estos deben recoger sus productos físicamente, pero hay dos diferencias principales y es que en Too Good To Go, es el establecimiento el que fija la hora de recogida y qué va dentro de cada lote de productos, mientras que en nuestra aplicación serán los clientes los que elijan tanto los productos como la hora de recogida dentro de unas franjas horarias delimitadas por los comercios.

Web	Pros
<b>Amazon</b>	Permite comprar rápidamente Elegir horario de entrega Elegir lugar de entrega Permite subir productos a vendedores Historial de pedidos
<b>TheFork</b>	Permite reservas por franja horaria Información detallada de cada restaurante
<b>TripAdvisor</b>	Permite realizar reservas en diversas actividades Valoración de los servicios ofrecidos
<b>Alibaba</b>	Ofrece exposición global a vendedores Permite a los usuarios comprar al por mayor Mayor libertad para vendedores para gestionar pedidos
<b>ToGoodToGo</b>	Permite reservar productos Los comercios eligen la hora de recogida Los usuarios no pagan hasta que recibir el pedido Los comercios pueden subir varios productos

**Tabla 2.1:** Esta tabla compara los pros de las distintas webs analizadas



Figura 2.6: Listado de packs en Too Good To Go

Web	Contras
Amazon	Necesario registro Menor autonomía para vendedores Distintos beneficios según tipos de usuario
TheFork	Necesario registrarse
TripAdvisor	Necesario registrarse Muestra menos información que otros servicios similares Necesario pagar previamente para algunos tipos de reservas
Alibaba	Necesario el registro Los usuarios deben negociar el envío con el vendedor
ToGoodToGo	Necesario registrarse Los usuarios no pueden elegir horario de recogida El pago es a través de la app Contactar con los administradores para dar de alta un nuevo comercio

Tabla 2.2: Esta tabla compara los contras de las distintas webs analizadas



## 2.5. Conclusión del estado del arte

Observando la tabla 2.1 se ven características que trataremos de plasmar en nuestra aplicación:

- Permitir comprar rápidamente (Amazon)
- Elegir horario de entrega (Amazon)
- Permitir añadir productos nuevos y fijar un precio (TooGoodToGo y Amazon)
- Libertad para vendedores para la gestión de pedidos (Alibaba)
- Permitir reservas por franja horaria (TheFork)
- Los usuarios pueden subir varios productos (TooGoodToGo)

Además de estas características que son positivas de los comercios vistos, también se implementarán algunas otras para salvar algunos de sus aspectos negativos reflejados en la tabla 2.2:

- No será necesario el registro para realizar una reserva (Amazon, TheFork, TripAdvisor, Alibaba y TooGoodToGo)
- Se mostrará toda la información necesaria de los productos (TripAdvisor)
- Los usuarios podrán elegir su horario de recogida, dentro de unas franjas horarias que establecerá el comercio. (TooGoodToGo)



## ANÁLISIS

---

Para definir los objetivos de este proyecto, se celebró una reunión con la panadería agroecológica Panifiesto, el comercio en el que se pretende implantar esta aplicación. En esta reunión se establecieron las funcionalidades y las condiciones de seguridad, privacidad, portabilidad, usabilidad, capacidad y eficiencia que debía tener la aplicación.

Durante la conversación surgieron diferentes ideas respecto a los usuarios. En concreto: podrán hacer reservas sin necesidad de estar registrados, pero estos deberán aportar correo electrónico y teléfono para recibir el resumen de la reserva y poder ser notificados en caso de ser necesario. Los usuarios que estén registrados podrán consultar todas sus reservas realizadas desde su menú de usuario y no tendrán que añadir sus datos en cada pedido.

Por otro lado, también se habló de las necesidades que debería cubrir para la digitalización de la panadería. Concretamente: deberán poder administrar todos los panes, las reservas y los horarios de apertura. Además, la aplicación deberá ser accesible desde el mayor número de plataformas posibles, al menos mediante los navegadores web principales, siendo interesante poder tener aplicación para iOS y Android. Dicha aplicación también deberá tener una interfaz sencilla e intuitiva ya que los clientes de la panadería abarcan distintos rangos de edades. Finalmente, las reservas de pan podrían contar con la opción de pago tras recogida o previo pago.

### 3.1. Objetivos

El objetivo principal es dotar de una aplicación web a un pequeño comercio para que sus clientes puedan realizar sus compras online y acudir a recoger su pedido en tienda. Esta aplicación deberá ser accesible desde los principales sistemas operativos como iOS, Android y Windows.

## 3.2. Catálogo inicial de requisitos

A continuación, una vez analizadas las necesidades del cliente se especifican los requisitos del sistema identificados inicialmente y divididos en requisitos funcionales y requisitos no funcionales.

### 3.2.1. Requisitos funcionales

#### **RF(1) - Realizar reservas en red**

El sistema deberá permitir a los usuarios realizar reservas en red, añadiendo los productos que deseen y eligiendo la hora de recogida.

#### **RF(2) - Realizar reservas telefónicas o presenciales**

El sistema deberá permitir a los administradores incluir reservas que los clientes realicen de manera presencial o telefónica para agruparlas en un mismo lugar.

#### **RF(3) - Ver listado de todas las reservas**

El sistema deberá permitir a los administradores acceder al listado de todas las reservas realizadas.

#### **RF(4) - Ver detalles de reservas**

El sistema deberá permitir a los administradores ver los detalles de una reserva, estos detalles deberán incluir toda la información necesaria para su preparación y cobro. Esta información incluirá los panes seleccionados, cómo se prepararán esos panes, la cantidad, el día y hora de recogida, el identificador de la reserva, el precio total, el estado en el que está y la información de contacto del cliente.

#### **RF(5) - Editar reservas**

El sistema, además deberá permitir a los administradores editar las reservas para poder indicar si alguna se cancela o si ya se ha recogido.

#### **RF(6) - Añadir nuevos productos**

El sistema deberá permitir añadir productos nuevos a los administradores especificando nombre, tipo de producto, días de la semana que estará disponible, una imagen del producto y precio por unidad.

**RF(7) - Ver listado de productos**

El sistema deberá permitir ver un listado con los productos creados, permitir añadir nuevos productos o ver sus detalles. Esta lista deberá mostrar los principales detalles de los productos, como su nombre, su precio por unidad, el tipo y los días que está disponible.

**RF(8) - Modificar productos**

El sistema deberá permitir a los administradores editar los productos. Se podrá editar la misma información que se recoge a la hora de crear un producto nuevo.

**RF(9) - Eliminar productos**

El sistema deberá permitir a los administradores eliminar productos de su catálogo, esta eliminación provocará que no aparezcan en el listado de productos de administración, ni listados para su compra por parte de los clientes.

**RF(10) - Modificar horarios de reserva disponibles**

El sistema deberá permitir a los administradores modificar las franjas horarias para hacer reservas.

**RF(11) - Ver horarios de reserva actuales**

El sistema deberá mostrar a los administradores los horarios disponibles para las franjas horarias de mañana y tarde para todos los días de la semana, siempre que tengan una franja definida.

**RF(12) -Añadir productos como usuario al carrito de la compra**

El sistema deberá permitir al usuario añadir los productos que seleccione a su carrito de la compra. A la hora de añadir un producto, deberá poder elegir dentro de las opciones que los administradores hayan registrado a la hora de crear el producto. Esta opción deberá estar disponible tanto para clientes registrados como para clientes sin registrar.

**RF(13) - Ver detalles de la reserva en el carrito de la compra**

El sistema deberá permitir al usuario ver los detalles de los productos añadidos al carrito. Dichos detalles deberán contener las opciones seleccionadas por el usuario, el número de unidades, el total para cada producto y un precio total. Esta opción deberá estar disponible tanto para clientes registrados como para clientes sin registrar.

#### **RF(14) - Editar pedido en el carrito de la compra**

El sistema en el carrito deberá permitir realizar cambios sobre los productos que haya añadido previamente. Esta opción deberá estar disponible tanto para clientes registrados como para clientes sin registrar.

#### **RF(15) - Eliminar productos del carrito de la compra**

Estando en el carrito de la compra el sistema deberá permitir a los usuarios eliminar productos previamente añadidos. Esta opción deberá estar disponible tanto para clientes registrados como para clientes sin registrar.

#### **RF(16) - Elegir fecha y hora de recogida**

Para poder hacer la reserva, el sistema deberá permitir a los usuarios seleccionar un día y una hora de recogida válidas para el horario que los administradores permitan.

#### **RF(17) - Registrar usuarios nuevos**

El sistema permitirá registrar nuevos usuarios, para ello deberá solicitar los datos de contacto, su nombre y una contraseña.

#### **RF(18) - Modificar datos de contacto**

El sistema permitirá a los usuarios registrados actualizar sus datos de contacto.

#### **RF(19) - Modificar contraseña**

El sistema permitirá a los usuarios actualizar su contraseña en caso de no recordarla.

#### **RF(20) - Ver listado de productos**

El sistema mostrará el catálogo de productos tanto a usuarios registrados como a usuarios no registrados, permitiendo ser añadidos dichos productos al carrito de la compra. El sistema deberá controlar que los productos solicitados estén disponibles el día de la reserva.

#### **RF(21) -Realizar reservas**

El sistema deberá permitir realizar reservas para todo tipo de usuarios, pero para usuarios no registrados deberá pedir sus datos de contacto en cada pedido.

### **RF(22) - Consultar historial de reservas**

Para usuarios registrados y logeados, el sistema deberá mostrarles un historial con sus reservas previas.

### **RF(23) - Detalles de reservas**

El sistema deberá permitir ver al usuario los detalles de sus reservas anteriores.

## **3.2.2. Requisitos no funcionales**

Ya hemos visto las funciones que tendrá que tener implementadas el sistema, a continuación veremos los requisitos que deberá cumplir la aplicación para satisfacer las necesidades del negocio.

### **Seguridad**

- RNF(1) - Debido a que se manejarán datos personales, se deberán tratar estos datos de acuerdo a la Ley de Protección de Datos.
- RNF(2) - Al realizarse reservas deberemos garantizar que un mismo usuario no pueda realizar múltiples pedidos de manera fraudulenta.

### **Portabilidad**

- RNF(3) - La aplicación deberá ser accesible desde cualquier navegador de manera online.
- RNF(4) - La aplicación deberá ser accesible desde los principales sistemas operativos: Windows, iOS y Android.

### **Usabilidad**

- RNF(7) - La aplicación deberá contar con iconos y lenguaje sencillos, ya que será utilizada por un público de un amplio rango de edad.
- RNF(8) - La interfaz deberá ser a su vez sencilla y mostrar todos los datos necesarios para poder realizar reservas.

### **Eficiencia**

- RNF(9) - Velocidad de reserva, el usuario podrá completar su reserva con menos de 10 clicks.
- RNF(10) - Al ser destinada a un pequeño comercio no se espera un volumen demasiado elevado de usuarios simultáneamente, por tanto deberá soportar unos 1000 usuarios simultáneos.

## **3.2.3. Casos de uso**

Ahora veremos todas las funciones que pueden realizar los tres tipos de usuarios que existirán en nuestra aplicación: usuarios no registrados, usuarios registrados y Administrador. Ya que algunas

acciones pueden realizarlas los tres tipos, en la figura 3.1 puede observarse un cuarto tipo que engloba a los otros tres con el fin de simplificar el diagrama de casos de uso.

La figura 3.1 también agrupa los casos de uso por subsistemas que se explican a continuación.

### Subsistema de gestión de pedidos

Este subsistema recoge los cuatro casos de uso relacionados con la realización de un pedido como son:

- Ver un producto
- Hacer un pedido
- Cancelar un pedido
- Ver historial de pedidos

### Subsistema de gestión de usuarios

El subsistema de gestión de usuarios se encarga de la gestión de los usuarios y sus datos en la aplicación y recoge los siguientes casos de uso:

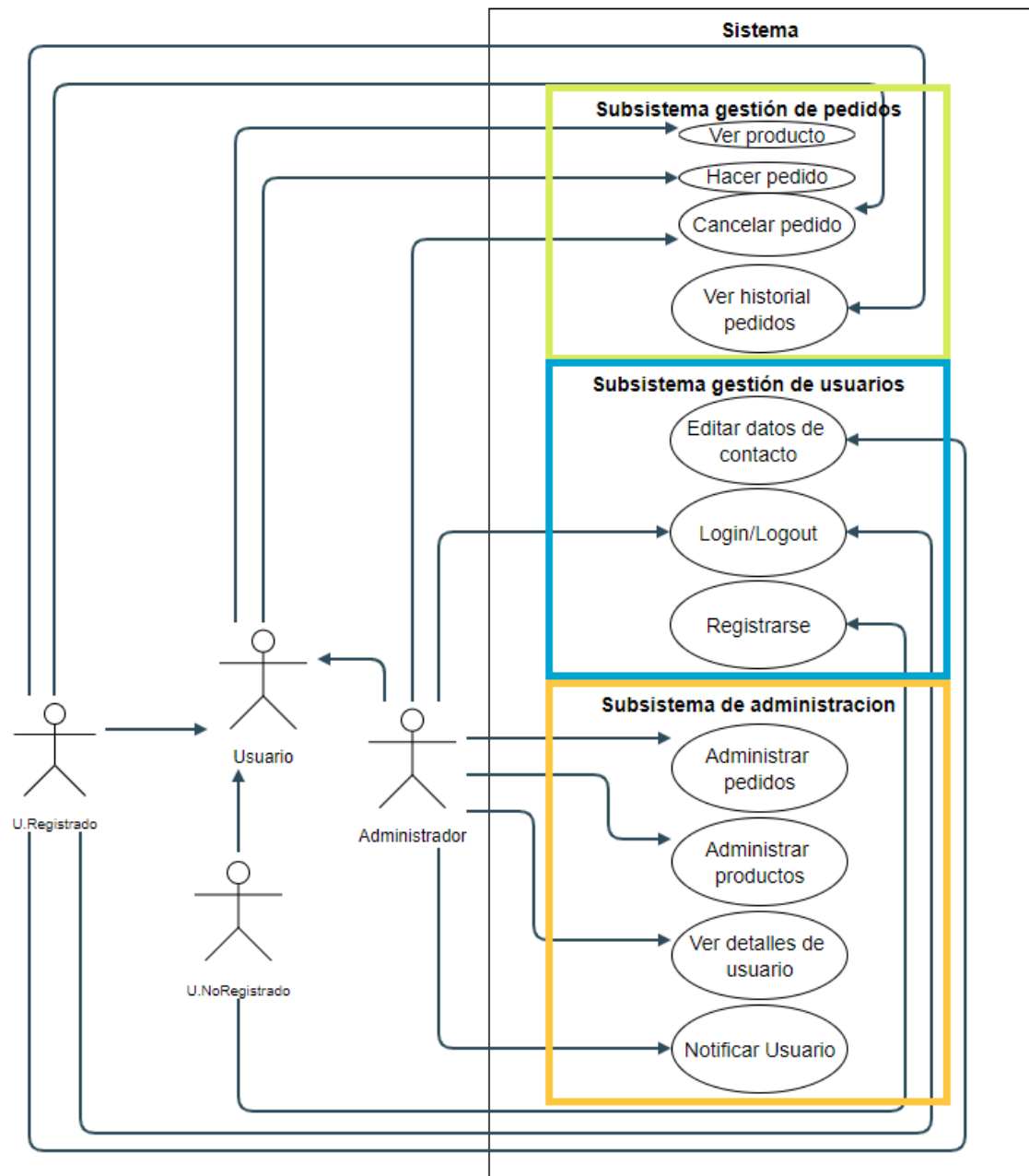
- Editar los datos de contacto
- Iniciar sesión en la aplicación
- Cerrar sesión en la aplicación
- Registrarse

### Subsistema de administración

Finalmente, el subsistema de administración engloba las funciones de los administradores de la aplicación, las cuales son:

- Administrar pedidos, ya sea añadiendo uno nuevo o editando un pedido previo.
- Administrar productos, permitir a los administradores la gestión de productos: añadirlos, editarlos o eliminarlos.
- Ver detalles del usuario, permitirá ver los detalles de contacto de los usuarios que realicen un pedido.
- Notificar a un usuario, con los datos de contacto podrán notificar a usuarios si hay algún problema con su pedido.



**Figura 3.1:** Casos de uso para todos los sistemas



# DISEÑO

---

En este punto se puede ver el diseño conceptual y visual de interacción del sistema, realizado con Moqups [1]. Ya que la aplicación será utilizada por un sector heterogéneo y de diversas edades, los cuales pueden no estar familiarizados con la tecnología, se ha buscado un diseño que resulte sencillo de interpretar y evite provocar confusiones. En los apartados anteriores se trata la descripción del proyecto, así como sus requisitos funcionales y no funcionales. Ahora es el momento de adentrarse esos requisitos de la aplicación mediante las siguientes maquetas.

En la figura 4.1(a) se puede apreciar el menú de navegación desplegado visto desde la perspectiva del usuario sin iniciar sesión. A continuación vamos a ver punto por punto cada parte de dicha pantalla:

- 1.— Este elemento indica al usuario que no ha iniciado sesión y tras ser pulsado, llevará al usuario a figura 4.2(a)
- 2.— Este enlace llevará a la página principal.
- 3.— Dirigirá a 'Quienes somos'.
- 4.— Pulsando sobre esta sección para acceder a la pantalla representada en la figura 4.3(a)
- 5.— Llevará a la figura 4.3(b).
- 6.— Pulsando se podrá ver el manifiesto en el que se basa el negocio.
- 7.— Pulsando aparecerá u ocultará el menú de navegación.
- 8.— Pulsando en la cesta, nos llevará a la sección de la figura 4.4(a) si el usuario está logeado o a la figura 4.4(b) en caso contrario.

Ahora se explicará la figura 4.2(a), esta es la pantalla para iniciar sesión. El usuario deberá introducir correo y contraseña. Además se pueden observar los siguientes elementos:

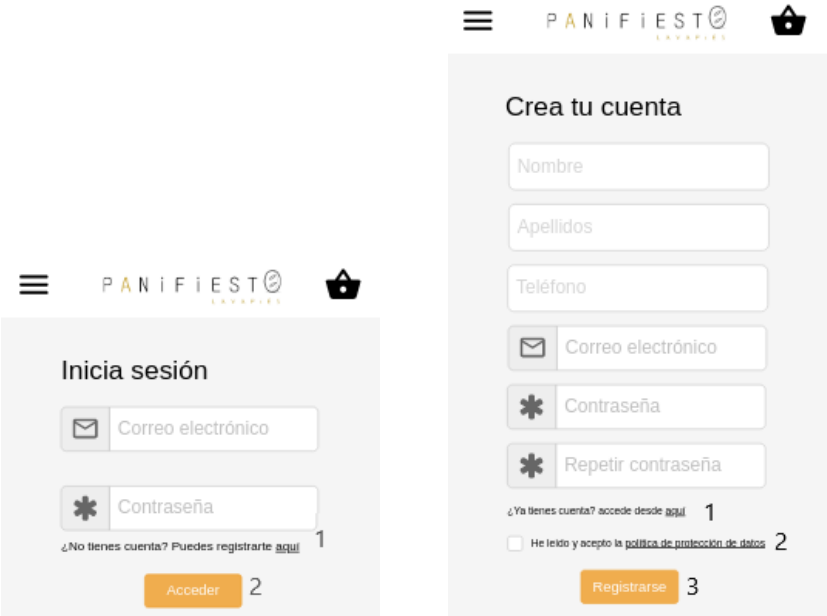
- 1.— Pulsando aquí llevará al usuario a la pantalla de registro, representada en la figura 4.2(b).
- 2.— Tras introducir su correo y contraseña, el usuario deberá pulsar este botón. En caso de acceso correcto se le redirigirá al inicio y el menú tendrá la apariencia de la figura 4.1(b)

A continuación se muestra la pantalla de registro representada en la figura 4.2(b), en ella el usuario deberá introducir su nombre, apellidos, teléfono y una cuenta de correo, así como una contraseña que utilizará para entrar en el sistema. Los datos personales como el nombre, apellidos, teléfono y correo electrónico, serán utilizados únicamente para comunicar cambios en sus pedidos así como para enviarles, en el caso del correo electrónico, un resumen de su pedido tras realizarlo. El resto de elementos que aparecen son los siguientes:



(a) Menu para usuarios sin logear (b) Menu para usuarios que iniciaron sesión

Figura 4.1: Menú de navegación de usuarios



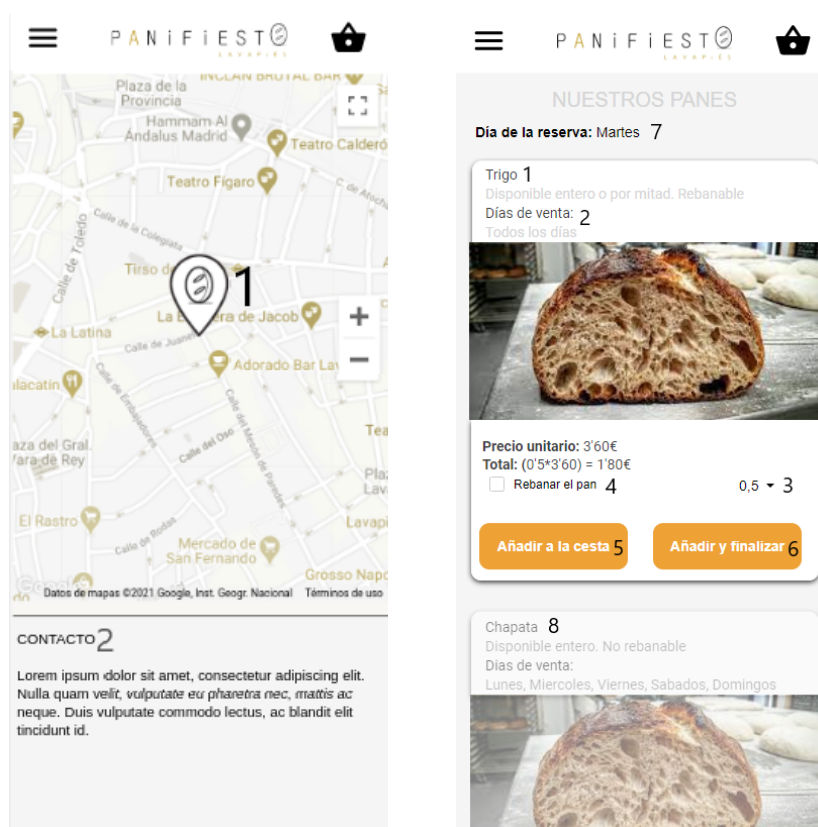
(a) Maqueta de pantalla de login (b) Maqueta de pantalla de registro

Figura 4.2: Pantallas de control de usuarios

- 1.— Permite al usuario volver a la pantalla de login en caso de haber pulsado por error.
- 2.— Este elemento deberá ser marcado por el usuario indicando que acepta la política de protección de datos, cuyo enlace llevará a un texto en el que se informe del uso de los datos que proporciona el usuario.
- 3.— Finalmente, tras pulsar este botón se verificarán que todos los campos estén completos y en caso afirmativo registrará al usuario y lo llevará a la pantalla de inicio.

En la figura 4.1(b) ya se puede ver el menú con un usuario que ha iniciado sesión. El único elemento diferente es:

- 1.— Este elemento al tener un usuario conectado, redirigirá al usuario a su menú representado en la figura 4.5.



(a) Maqueta de pantalla donde estamos

(b) Maqueta de pantalla de nuestros panes

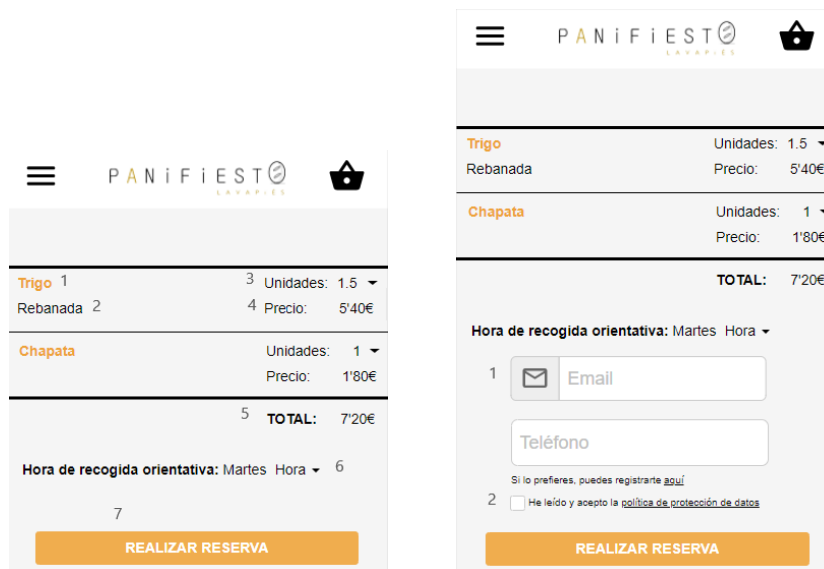
**Figura 4.3:** [Pantallas de contacto y catálogo de productos

Observando la figura 4.3(a) se puede ver que se trata de la sección de contacto, cuyos elementos son:

- 1.— Este elemento es el marcador en el mapa e indica la posición de la tienda.
- 2.— Por otra parte, este elemento tendrá los datos de contacto como la dirección, teléfono, correo electrónico y horario de apertura.

A continuación se observa la figura 4.3(b). Se trata de una de las pantallas con mayor funcionalidad de la aplicación, ya que muestra el catálogo de panes. Esta pantalla consta de varios elementos que procedo a explicar:

- 1.— Tarjeta que muestra toda la información del pan, permite seleccionar cantidad y añadir a la cesta o añadir a la cesta y finalizar.
- 2.— Aquí se muestra la información del pan, nombre, si se puede pedir por mitades, si es rebanable y los días que está a la venta.
- 3.— Aquí hay un selector de unidades, depende del tipo de pan aceptará solo enteros o también permitirá mitades.
- 4.— Seleccionando este elemento, se indica que se desea el pan rebanado.
- 5.— Pulsando este botón, se añadirá el pan a la cesta y permitirá seguir navegando por la pantalla.
- 6.— Pulsando añadirá el pan a la cesta y llevará al usuario a la cesta, representada en las figuras 4.4(a) y 4.4(b).
- 7.— Este elemento deberá ser seleccionado cuando el usuario acceda a la sección. Si el usuario no selecciona, aparecerán todos los panes sin la opción de ser añadidos, ya que no todos los panes estarán disponibles a diario. Una vez seleccione el día de la reserva, se mostrarán los panes disponibles en primer lugar.
- 8.— Finalmente, este elemento representa un pan no disponible, el cual será visible, pero no permitirá ser añadido a la cesta porque no está disponible el día que el usuario ha elegido para la reserva.



(a) Maqueta de la pantalla de cesta para usuarios registrados

(b) Maqueta de la pantalla de cesta para usuarios no registrados

**Figura 4.4:** Pantalla de cesta

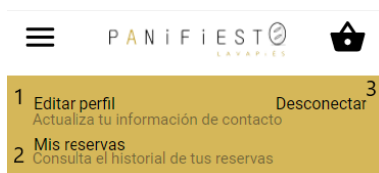
Ahora hablaré de la cesta representada en 4.4(a), la cual muestra un resumen del estado actual del pedido y permite editar distintos elementos:

- 1.— Nombre del producto
- 2.— Si se quiere rebanada o no rebanada, en caso de ser un pan rebanable.
- 3.— Representa el número de unidades seleccionadas, podrá ser actualizado.
- 4.— Muestra el precio total del producto respecto al número de unidades seleccionadas, en caso de cambiar las unidades, se actualizará el precio.
- 5.— Es la suma total de todos los productos de la cesta.
- 6.— Este selector, permite elegir la hora orientativa de recogida.

7.— Finalmente, el usuario pulsará este botón para realizar su reserva, lo cual le llevará a la página principal.

La siguiente maqueta a describir es la de la figura 4.4(b), la cual tiene las mismas funcionalidades que la figura 4.4(a), solo que solicitará al usuario sus datos de contacto:

- 1.— El usuario deberá proporcionar email y teléfono. El primero será necesario para recibir el resumen del pedido y el segundo, será necesario por si hubiese algún problema con el pedido pudiesen ponerse en contacto desde la panadería.
- 2.— El usuario sin registrar deberá confirmar cada vez que haga un pedido sus datos y aceptar el tratamiento de los mismos.



**Figura 4.5:** Maqueta de la pantalla de menú de usuario

La figura 4.5 es el menú que ven los usuarios registrados, el menú cuenta con estos elementos:

- 1.— Pulsando aquí, el usuario accederá a la pantalla de la figura 4.8.
- 2.— Por aquí acceden a su historial de reservas, representado en la figura 4.6.



**Figura 4.6:** Maqueta de la pantalla del historial

A continuación, viendo la figura 4.6 que muestra el listado de los pedidos realizados, tiene dos elementos a destacar:

- 1.— Este hace referencia a los datos que se muestran, el historial mostrará los datos de día, hora y el total.
- 2.— Si el usuario pulsa sobre cualquier elemento de la lista, se abrirá una pantalla con los detalles del pedido en concreto, representado en la figura 4.7.

La figura 4.7 representa los detalles de un pedido, estos detalles son similares a la pantalla del carrito, pero no permitirán su edición, puesto que es una pantalla meramente informativa. Los elementos señalados son:

- 1.— Este elemento, muestra la información del pan que se reservó, cómo se quería, en este caso rebanado, el número de unidades y cuanto costaron esas unidades en total.
- 2.— Con este, vemos el día de la semana y la hora orientativa a la que hicimos la reserva.
- 3.— Finalmente, si pulsan en cualquier parte del elemento, serán redirigidos a la pantalla de 'Nuestros panes', figura 4.3(b).



Maqueta de pantalla de detalles del pedido. La interfaz muestra el logo 'PANIFIEST' y un ícono de carrito de compras. El pedido está detallado en una tabla:

Trigo 1	Unidades: 1.5
Rebanado	Precio: 5'40€
Chapata	Unidades: 1
3	Precio: 1'80€
<b>TOTAL: 7'20€</b>	

En la parte inferior, se indica: **Hora de recogida orientativa:** Martes 15:30 <sup>2</sup>

**Figura 4.7:** Maqueta de pantalla de detalles del pedido



Maqueta de pantalla para actualizar datos. La interfaz muestra el logo 'PANIFIEST' y un ícono de carrito de compras. El título es 'Actualiza tus datos'. Hay dos campos de entrada:

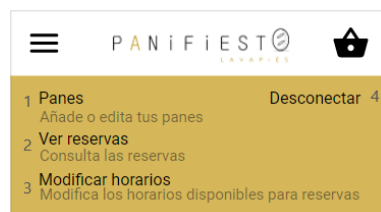
- 1 Teléfono
- 2 Correo electrónico

Debajo de los campos hay un botón naranja que dice 'Actualizar datos' con el número 3 a su lado.

**Figura 4.8:** Maqueta de pantalla para actualizar datos

La siguiente figura es la 4.8. Esta figura muestra los campos que el usuario puede actualizar desde este menú, los cuales serán los datos de contacto. No hay ningún problema en cambiar estos datos, ya que los usuarios tendrán un índice único asociado a su cuenta, lo que permitirá conservar todos sus pedidos aun que el correo o el teléfono asociados cambien. Los elementos diferenciados en esta pantalla son 3:

- 1.— El teléfono permite ser cambiado ya que es un método de contacto que puede ser susceptible a cambios por parte del usuario. Dicha modificación no afecta al funcionamiento de la aplicación.
- 2.— El correo electrónico, puede ser necesario cambiarlo por múltiples razones. Por ello, se enfoca la aplicación para evitar que el cambio de correo electrónico suponga una pérdida de los pedidos del usuario.
- 3.— El elemento actualizar datos, comprobará que ambos campos estén rellenos y sean correctos. En caso de no rellenar ambos campos, tan solo se actualizará el que se haya rellenado siempre que este tenga una estructura correcta.



Maqueta de la pantalla del menú de administrador. La interfaz muestra el logo 'PANIFIEST' y un ícono de carrito de compras. El menú está detallado en una tabla:

1 Panes	Desconectar 4
Añade o edita tus panes	
2 Ver reservas	Consulta las reservas
3 Modificar horarios	Modifica los horarios disponibles para reservas

**Figura 4.9:** Maqueta de la pantalla del menú de administrador



A continuación menú de administración representado en la figura 4.9, el cual solo podrá ser accesible mediante una cuenta de administración que será proporcionada a la panadería. Esta contiene 4 elementos destacables:

- 1.— El elemento 'Panés' llevará al menú de panes de la figura 4.10.
- 2.— El elemento 'Ver reservas' llevará al menú de reservas de la figura 4.12.
- 3.— El elemento de 'Modificar horarios' llevará al usuario a la figura 4.14.
- 4.— 'Desconectar' cerrará la sesión de administrador y llevará al usuario a la pantalla de inicio.



**Figura 4.10:** Maqueta de la pantalla de panes

La siguiente maqueta a explicar es la de la figura 4.10, la pantalla de panes. Esta pantalla permitirá a los administradores añadir y editar los panes y la información de los mismos que luego aparecerá en la figura 4.3(b). Los elementos a destacar son:

- 1.— El botón de añadir, como su nombre indica, permitirá añadir nuevos panes al catálogo de panes existentes de la aplicación. Para esto llevará al administrador a la pantalla de la figura 4.11(a).
- 2.— El botón de editar, llevará a la pantalla de la figura 4.11(b)
- 3.— Además del botón de editar, también producirá el mismo efecto el pulsar sobre los detalles de los panes listados.

La pantalla de creación de pan es la siguiente, aparece representada en la figura 4.11(a). Esta pantalla consta de todos los elementos necesarios para añadir toda la información necesaria para poder reservar correctamente el pan por parte de los usuarios. Esos elementos son los siguientes:

- 1.— El nombre del pan, no tiene porque ser único ya que internamente utilizarán índices únicos asociados a cada pan.
- 2.— Este campo será el precio del pan, es un campo numérico y representará su valor por unidad.
- 3.— Marcar esta casilla indicará que el pan podrá ser solicitado por el usuario en formato rebanado.
- 4.— Si el administrador marca esta casilla, permitirá que los usuarios puedan comprar dicho pan en unidades enteras o por mitades.
- 5.— El nombre de la imagen a subir por parte del usuario.
- 6.— Este botón, cuando el administrador pulse, le aparecerá la opción de subir una imagen almacenada en su dispositivo.
- 7.— Los días de la semana, deberán ser marcados aquellos en los que el pan vaya a estar disponible. En el ejemplo de la figura 4.11(a) los martes, jueves y viernes, no estaría disponible este pan. Aun que no esté disponible esos días, si aparecerá en el listado de todos los panes, con la diferencia de no poder ser añadido al carro.
- 8.— Además de elegir los días, también damos la opción de tener el pan disponible u ocultarlo. Esta funciona-

lidad puede ser necesaria si se quiere añadir un pan durante una fecha del año determinada, luego podría ser ocultado sin necesidad de eliminarlo. Esto permitiría que no se muestre en los panes, pero siga estando presente para cuando se quiera mostrar de nuevo, si necesidad de cambiar todos los datos de nuevo.

9.— Aparecerá deshabilitado hasta que se completen todos los campos anteriores. Una vez esten completos, se activará y permitirá añadir el producto.

(a) Pantalla de creación de pan

(b) Pantalla de edición de pan

**Figura 4.11:** Pantallas de gestión de panes

La figura 4.11(b) muestra la pantalla de editar el pan que es similar a la de creación solo que con dos elementos extra:

- 1.— Este elemento, guarda los cambios que se hagan en la información del pan
- 2.— Este, permite eliminar el pan en caso de ser necesario.

**Figura 4.12:** Maqueta de pantalla del listado de reservas

En la figura 4.12 puede verse el listado de reservas. Este listado contiene los siguientes elementos:

- 1.— Columna que muestra el número de reserva, será único para cada reserva.
- 2.— Esta columna, muestra la hora aproximada sobre la que pasarán a recoger el pedido.
- 3.— Coste total del pedido.
- 4.— Indica el estado del pedido, podrá ser Recogido, Pendiente o Cancelado.
- 5.— Este botón, abrirá la figura 4.13 donde permite añadir un pedido nuevo manualmente.
- 6.— Pulsando este botón, abrirá la figura 4.14 que muestra los detalles del pedido.
- 7.— Este elemento indica para que día son los pedidos, aparecerán agrupados por día.
- 8.— Finalmente, este elemento, permitirá buscar por el identificador del pedido.

1

Trigo ☐ Rebanar Unidades: 1.5 ▼

Chapata Unidades: 0 ▼

Gallego ☐ Rebanar Unidades: 0.5 ▼

Total: 7.35€

Hora de recogida orientativa: 3 Martes ▼ Hora ▼ 2

4 Teléfono

5 Nombre

6 REALIZAR RESERVA

**Figura 4.13:** Maqueta de pantalla de añadir reserva manualmente

La pantalla representada en la figura 4.13 permitirá a los administradores crear pedidos, lo que permitirá seguir recibiendo pedidos de manera telefónica y registrarlos en el sistema. Esta pantalla tiene los siguientes elementos:

- 1.— Lista con todos los panes, disponibles o no en función de la fecha que se escoja en 3, además permitirá seleccionar las unidades y si se quiere rebanado o no, mostrando el precio final.
- 2.— Permite seleccionar la hora aproximada a la que se recogerá.
- 3.— Este elemento, al igual que en la sección de la figura 4.3(b), permitirá elegir el día que se recogerá la reserva. En función del día que se escoja, los elementos no serán seleccionables en el listado superior.
- 4.— Teléfono del cliente, por si fuese necesario comunicarse con el.
- 5.— Nombre, para tener un nombre asociado al pedido.
- 6.— Finalmente, pulsando aquí se generará una nueva reserva con los datos de contacto proporcionados, este botón permanecerá inhabilitado mientras que no haya datos de contacto, fecha y hora de recogida y al menos un pan seleccionado.

Por otro lado, los detalles del pedido representados en la figura 4.14, solo será modificable el estado del pedido. Destaco los siguientes elementos:

- 1.— Resumen con la información más importante del pedido y los datos de contacto.

Número de pedido: XDK453  
 Hora: 14:00  
 Total 3'60€  
 Nombre cliente: Ander  
 Teléfono: 666666666

2 Recogido

Recogido

Pendiente

Cancelado

Chapata 3 Unidades: 2 Precio: 3,60€

TOTAL: 3,60€

4 Guardar

5 Eliminar

**Figura 4.14:** Maqueta de pantalla de detalles de la reserva

- 2.– Este selector, permitirá editar el estado del pedido entre los 3 posibles.
- 3.– En esta parte, aparecerá el listado completo de productos, sus unidades, precio y el precio total.
- 4.– Guarda los cambios del pedido.
- 5.– Elimina el pedido, puede ser útil en caso de generar un pedido por error.

Finalmente, en la figura 4.15, se muestran los horarios de recogida disponibles en función de los días de la semana. Los elementos a destacar son los siguientes:

- 1.– Este selector, será un selector múltiple, que tendrá como opciones todos los días de la semana.
- 2.– Hora de inicio de reservas por la mañana.
- 3.– hora de final de reservas por la mañana.
- 4.– Hora de inicio de reservas por la tarde.
- 5.– Hora de final de reservas por la tarde.
- 6.– Tabla que muestra las franjas de recogida disponibles para que los usuarios puedan recoger su pedido.
- 7.– El botón guardar, comprobará los campos de 2 al 5 y actualizará esos datos para todos los días que hayan sido seleccionados en 1. En caso de no tener hora de apertura y cierre para por la mañana o por la tarde, se actualizará su valor y se quitará esa opción de las horas seleccionables para el usuario.

PANIFIEST

1

2

3

4

5

Días ▾

10:30

14:30

18:30

19:30

	Mañana		Tarde	
Lunes	10:30	14:30	18:30	19:30
Martes	10:30	14:30	18:30	19:30
Miercoles	10:30	14:30	18:30	19:30
Jueves	10:30	14:30	18:30	19:30
Viernes	10:30	14:30	18:30	19:30
Sabados	10:30	14:00		
Domingo	10:30	14:00		6

7

Guardar

Figura 4.15: Maqueta de pantalla de edición de horarios de reserva



# IMPLEMENTACIÓN

---

Vistos ya el análisis, extraídos los objetivos, los requisitos y realizados los diseños conceptuales de la aplicación, toca centrarse en la implementación.

Para elegir que framework y lenguaje utilizar, se realizó una comparativa entre algunos los distintos lenguajes y frameworks que permitirían realizar una aplicación web que a su vez pudiese ser instalada en terminales móviles. A continuación se procede a exponer cada una de las distintas opciones y finalmente cuales elegiremos y el por qué.

## 5.1. Python

Python es una de las primeras opciones que se tuvieron en cuenta, es un lenguaje de programación de alto nivel que puede ser utilizado para la programación de aplicaciones web. Pero investigando más a fondo, no muchos hosting admiten python como lenguaje de programación, lo cual es un problema.

## 5.2. React y Angular

Tras proseguir la investigación de otras tecnologías para el desarrollo web, nos encontramos con React y Angular [2], ambos son un framework de javascript. Estos frameworks están más extendidos que el de Python y son aceptados en gran parte de los hosting.

## 5.3. App nativa o Progressive Web App

Ya que uno de los requisitos que se busca en la aplicación es que sea poder ser utilizada en dispositivos móviles. Para dotar al comercio de su propia aplicación había opciones de generar una app nativa para cada sistema y subirla al mercado de aplicaciones de cada sistema operativo, o crear una Progressive Web App (PWA) que pueda ser instalada en la pantalla de móvil del usuario como si de una app nativa se tratase.

Una aplicación nativa implica gastos para el comercio, puesto que tendrá que pagar para poder añadir su aplicación a las tiendas, así como diferentes requisitos que deberá cumplir para superar las verificaciones de ambas tiendas, por otro lado resulta más fácil acceder a los recursos del terminal como pueden ser el GPS o la cámara. Una PWA por su parte, evita los gastos y requisitos asociados a añadir la aplicación a las distintas tiendas, pero tiene limitaciones respecto al uso de recursos del terminal.

Ionic es un framework que permite trabajar tanto con PWA como con Apps nativas facilitando el desarrollo de aplicaciones web y su adaptación a dispositivos móviles. Dicho framework se integra perfectamente tanto Angular como React.

## 5.4. Firebase

A la hora de gestionar los datos, su almacenamiento y su utilización se utilizará Firebase, que es una herramienta en la nube perteneciente a Google, la cual nos permite almacenar datos, gestionar usuarios y contraseñas todo haciendo uso de su API la cual es completamente compatible con las tecnologías JavaScript mencionadas anteriormente en los apartados 5.2 y 5.3.

## 5.5. Javascript vs Typescript

JavaScript es un lenguaje de programación creado para mejorar la navegación para el usuario desde el lado del navegador. Hemos visto en la sección 5.2 que tanto Angular como React son frameworks para JavaScript.

Por otro lado, tenemos TypeScript, una mejora del lenguaje JavaScript. Con TypeScript podemos realizar las mismas funciones que con JavaScript pero evitando errores producidos por la falta de tipado de JavaScript. Teniendo en cuenta que TypeScript nos ofrece lo mismo que JavaScript añadiendo tipos tanto para datos como para funciones y viendo que también es compatible con Ionic, React y Angular, será la mejor opción entre ambos.

## 5.6. Tecnologías a utilizar

Una vez vistas varias tecnologías es el momento por decantarnos por las que se emplearan en el desarrollo del proyecto. Descartaremos Python porque es una tecnología que nos podría dar problemas a la hora de pasar la versión a producción y alojarla en un hosting. Nos decantaremos por el uso del framework de Ionic, el cual nos permitirá trabajar con React o con Angular. Tanto Angular como React son perfectamente compatibles para crear la aplicación, pero nos decantaremos por React, por



aprender más de este framework. Por tanto, para la parte frontal de la aplicación elegiremos Ionic-React haciendo uso de TypeScript por lo comentado en la sección 5.5.

Para la parte de backend, utilizaremos Firebase, en concreto su herramienta de Realtime Database para el almacenamiento de datos y Authentication para la gestión de usuarios. Lo cual nos facilitará el trabajo haciendo uso de su API, la cual está adaptada para trabajar con React como se menciona en 5.4

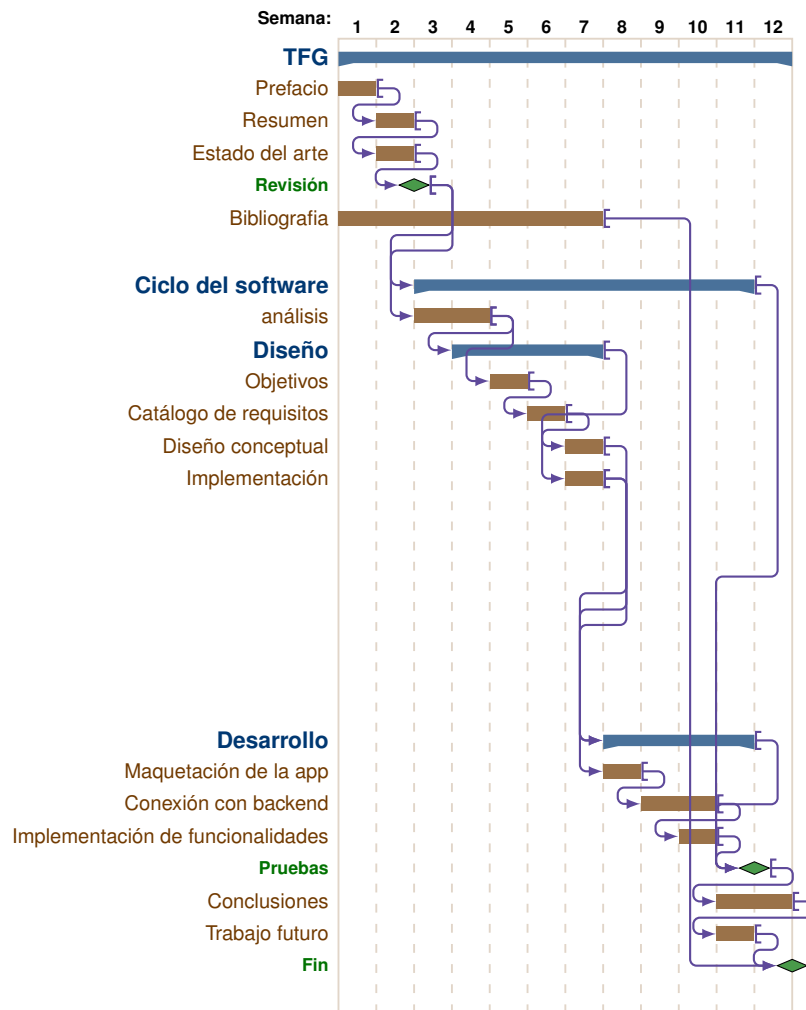
## 5.7. Ciclo de vida del software

Una vez decididas las tecnologías que se utilizarán para gestionar tanto la parte de frontal como la parte de backend, es momento de decidir el ciclo de vida del software.

Teniendo en cuenta que el equipo de trabajo está compuesto tan solo por una persona, convenirá utilizar un modelo iterativo, el cual permitirá corregir fallos que puedan haber surgido durante el desarrollo, o implementar nuevas funcionalidades. Tras cada iteración tendremos un prototipo con las funcionalidades que se hayan establecido previamente. Para cada prototipo se establecerán unos objetivos y unos casos de prueba, se comenzará el desarrollo. Tras acabar el desarrollo el prototipo se llevará una fase de comprobación en la que se verificará el correcto funcionamiento de las nuevas funcionalidades así como de las anteriores, en caso de que puedan haberse visto afectadas. Como podemos observar en la figura 5.1 está estipulada una duración de tres meses, esta estimación tiene en cuenta dejar un prototipo que permita añadir nuevos productos y consultar pedidos a los administradores y que permita reservar y pagar a los clientes.

Además de todo lo mencionado anteriormente, utilizaremos la tecnología Git para el mantenimiento de versiones. Al tener un control de versiones será mucho más sencillo volver a un punto anterior del código en caso de error y nos permitirá evaluar nuevas funcionalidades sin afectar a la versión principal de la aplicación.

En nuestro caso utilizaremos Bitbucket, pero podría haberse elegido otra ya que Ionic está adaptado para permitir generar las aplicaciones desde cualquier solución Git.



**Figura 5.1:** Diagrama que muestra el tiempo estimado del desarrollo y su evolución

## PRUEBAS

---

Todo desarrollo debe tener su fase de pruebas en el que se compruebe que todo tiene el funcionamiento que se esperaba. Para poder verificar que funciona correctamente nuestra aplicación realizaremos pruebas unitarias para cada componente en las que comprobemos que tiene el comportamiento esperado establecido previamente en el análisis.

También, cómo hemos visto en el capítulo 5 sección 5.7, realizaremos pruebas al finalizar la parte del desarrollo. Dichas pruebas consistirán en comprobar los casos de uso vistos en la sección 3.2.3 y que los requisitos funcionales planteados en la sección 3.2.1 se satisfacen.

Finalmente, tras finalizar las pruebas anteriores, ofrecimos probar el prototipo de la aplicación a diferentes usuarios de diferentes rangos de edad, los cuales nos describieron su experiencia con la aplicación, así como qué cosas echaron en falta.

Al entrar en la aplicación los usuarios echan en falta una página de presentación del comercio, comentando quienes son y como realizan sus productos, ya que algunos de ellos no conocen el lugar y no saben las diferencias con otros comercios de la misma venta de productos.

Al realizar la reserva de panes, comentan que les es muy cómodo la visualización del producto por imagen y también su disponibilidad, ya que no tienen que esperar a realizar la reserva entera para ver si esos días el producto está a la venta.

También sugieren, al añadir diferentes productos, en el carrito se mostrara cuantos productos se están pidiendo mientras ven los diferentes panes. Al finalizar la compra, al no estar registrado, ven lógico pedir datos de teléfono y correo, ya que de esta forma pueden coincidir nombres, pero se diferencian en los demás datos, pero creen que, al darle a solicitar reserva, debería de enviarse un correo de confirmación, para que el cliente tuviera la tranquilidad de que al llegar a recoger su producto, el vendedor tuviera forma de verificar que es dicha reserva.

Los usuarios prefieren, o ven más atractivo el que aparezca antes de pedir (ya sea con usuario o sin él), la pantalla de registro, ya que de está forma se animarían más clientes a registrarse, porque no tienen que ir a ninguna pagina distinta. Si no quisieran, se podría añadir a la pantalla de registro una pestaña de realizar pedido sin registrar para que se tuviera la opción de realizar la reserva sin

contraseña.

Cuando los clientes utilizan la aplicación siendo usuarios, les gusta ahorrarse tiempo en registrar datos y también el ver sus pedidos anteriores, ya que les resulta más fácil repetir reserva y ahorrar tiempo en buscar los productos.

## CONCLUSIONES

---

En este trabajo de fin de grafo se ha desarrollado un sistema informático para la gestión de reservas y pago en remoto, con un catálogo de productos que permite ser renovado sin nuevas implementaciones.

A nivel técnico, esto supuso una investigación previa de las tecnologías de desarrollo web existentes para poder elegir las que mejor se adaptaban para implementar un sistema con modelo cliente-servidor. Se investigaron previamente las más comunes en programación web como son HTML, PHP además de las que se especifican en el capítulo 5, como son Python, JavaScript, Ionic, React [3] y Angular, más orientadas a la parte del cliente de la aplicación y Firebase más enfocado a la parte del servidor. También, para alojar las imágenes que se utilizarán para los productos, hubp que investigar sobre los Content Delivery Network, red de distribución de contenidos (CDN) [4].

También supuso extraer los requisitos de una conversación con un usuario real y saber darles forma. Para ello, repasé la conversación varias veces, anotando las características que iba extrayendo cada vez. Una vez extraídas las funcionalidades requeridas, amplié su definición para que a la hora de implementarlo no pudiese haber ambigüedades. Tras tener los requisitos separados por funcionales y no funcionales, trabajé en los diseños mediante maquetas. Tener unas maquetas sobre como han de ser las diferentes pantallas de nuestro sistema facilitó el posterior desarrollo ya que había unos diseños en los que basarse para estructurar las cosas, proporcionando una línea gráfica similar para toda la aplicación.

Una vez vistas las distintas tecnologías y extraídas las funcionalidades hubo que elegir cuales se emplearían para realizar la funcionalidad. Para la parte frontal de la aplicación y al haber descartado previamente Python por dificultades en escalabilidad, tenía que elegir entre React o Angular, entre estas dos la decisión era arbitraria y me decanté por React que tan solo lo había visto por encima y tras ver que Firebase encajaba perfectamente en lo que la aplicación necesitaba la elegí como herramienta para el servidor de la aplicación, concretamente Realtime Database [5] y el uso de Google Drive como CDN.

La elección de React me permitió aprender enormemente sobre el framework, amplié mi conocimiento sobre los hooks y sus usos. También a realizar un modulo de conexión con un API y el uso

de los contextos [6] [7] para el traspaso de información entre distintos componentes de una aplicación web.

El servidor al ser implementado mediante un API de terceros implicó leer la documentación e investigar algunos usos concretos a parte [8] [9], ya que no aparecían en la documentación. Esto hizo que dominase con mayor soltura la utilización del API de Firebase [10], con el que ya había trabajado durante la carrera, pero en menor profundidad.

Para mí ha sido un gran oportunidad de aprendizaje en todos los sentidos, ya que me ha demostrado que soy capaz contar para el desarrollo con la participación de un usuario real, recopilando la información que me ofrece y transformándola en requisitos necesarios para el sistema, confeccionando las maquetas y aprendiendo mientras avanzaba sobre las tecnologías que estaba empleando. Además de llevar el desarrollo de la aplicación en solitario, permitiéndome explorar todas las áreas del desarrollo software y ayudándome a ver la importancia de todas y cada una de ellas para que todo funcione correctamente.

## TRABAJO FUTURO

---

Una vez hecho el desarrollo y las pruebas, han quedado incompletos algunos de los requisitos del cliente directo. Estos requisitos serán completados antes de la puesta en producción y se detallan a continuación los motivos por los que no son posibles a día de hoy.

Uno de ellos es el de permitir pagar la reserva desde la plataforma. Este requisito está contemplado en el flujo pero no está implementado ya que al involucrar transacciones bancarias, será necesario que el cliente contrate dicho servicio previamente. Una vez contratado, se deberá unir la entrada de la API que el cliente haya contratado y su salida para completar el flujo del pago.

Otro de los requisitos que ha quedado pendiente es el de añadir un mapa en el que aparezca el establecimiento marcado, este requiere dar de alta una cuenta de desarrollador en Google Cloud para obtener un identificador de uso de la API. Para conseguir esto, es necesario dar de alta una cuenta con método de pago y podría incurrir en costes, por ello se hablará con el cliente para valorar opciones.

Por otro lado, algunas funcionalidades no requeridas pero que podrían ser implementadas en el futuro son las siguientes: Aprovechando los datos obtenidos de compras implementar un sistema de aprendizaje automático con el fin de ayudar en la planificación y gestión del comercio. Como sugirieron algunos de los usuarios que pudieron probar la aplicación, sería interesante contar con un sistema que envíe una confirmación automáticamente de la reserva. También otra funcionalidad importante sería el añadido de un elemento visual que confirme que los artículos se van añadiendo al carrito. Además el sistema de notificación podría incluir recordatorios en el día de la reserva y en un periodo anterior a la hora fijada por el cliente. Esto último facilitaría mucho a los comercios ya que no tendrían que hacer recordatorios de manera telefónica, dejándoles más tiempo para la atención a los clientes y la preparación de las reservas.

Asimismo el trabajo futuro incluirá labores de mantenimiento ante posibles errores que no hayan sido detectados en las pruebas y actualizaciones del sistema para evitar fallos derivados del uso de versiones obsoletas.





# BIBLIOGRAFÍA

---

- [1] "Moqups." [Online; consultado el 12-Mayo-2021] (*Consultar*).
- [2] "Angular vs react – a complete comparison (2021)." [Online; consultado el 15-Mayo-2021] (*Consultar*).
- [3] "React-bootstrap documentation." [Online; consultado el 14-Junio-2021] (*Consultar*).
- [4] "How to use google drive as cdn for your website." [Online; consultado el 23-Mayo-2021] (*Consultar*).
- [5] "React realtime database." [Online; consultado el 19-Mayo-2021] (*Consultar*).
- [6] "How to use context in react with hooks." [Online; consultado el 1-Junio-2021] (*Consultar*).
- [7] "Context." [Online; consultado el 30-Mayo-2021] (*Consultar*).
- [8] "Handling user authentication with firebase in your react apps." [Online; consultado el 21-Mayo-2021] (*Consultar*).
- [9] M. Ogbopina, "How to build an authentication system in react using firebase." [Online; consultado el 21-Mayo-2021] (*Consultar*).
- [10] S. Murthy, "Build a todo app using firebase + react + typescript." [Online; consultado el 17-Mayo-2021] (*Consultar*).



# ACRÓNIMOS

---

**CDN** Content Delivery Network, red de distribución de contenidos.





